Statistical Techniques in Robotics (16-831, S21) Lecture #12 (Monday, March 15)

Bandits (Explore-Exploit), UCB

Lecturer: Kris Kitani *Scribes: Abhinav Agarwalla, Kshitij Goel*

1 Review

In the previous lecture, we finished learning about AdaBoost algorithm and started understanding the Multi-Armed Bandit problem and one algorithm (Explore-Exploit) for it.

1.1 Adaboost

1.1.1 PAC Learning Model

The PAC learning model is a theoretical framework to answer two questions:

1. What is the optimal dataset size to obtain good generalization?
2. What is the computational cost of learning?

1.1.2 Boosting

The theorem based on [2] suggests that a weak PAC learner can be called multiple times to generate a sequence of hypothesis with a different subset of **D** which can be combined to form a single strong PAC learner. This process is called *boosting*.

Adaboost [1] is a specific learning algorithm in the class of boosting algorithms. In previous online learning methods, one receives new data points at every time instant. In Adaboost, we receive *weak learners* instead of data points. The final prediction rule is a weighted sum of learner predictions upto the current time T. Since we have T weak learner at time step T, we just use the 'number of weak learners' terminology over time steps. The Adaboost algorithm is detailed in Algorithm 1.

1.2 Multi-Armed Bandits

1. **One-Shot Feedback**: This is because one action from the player leads to one reward and selecting any particular action does not change the state at the next time-step.
2. **Exhaustive Feedback**: The player can pull all the arms for the duration of the game. The state and action spaces are finite.
3. **Evaluative Feedback**: The player receives a reward sampled from the underlying unknown reward distributions at each time-step.

Algorithm 1 Adaboost

Input: Dataset $\mathbf{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$, where feature vector $\mathbf{x}_n \in \mathbb{R}^M$ and label $y \in \{0, 1\}$
Input: Weight vector $\{w^{(0)}\}_{n=1}^N$ initialised uniformly
Input: Number of weak learners T
for $t = 1, \cdots, T$ **do**
  $\mathbf{p}^{(t)} = \mathbf{w}^{(t-1)} / (\sum_n w_n^{(t-1)})$
  $h^{(t)} = \texttt{WEAKLEARNER}(\mathbf{D}, \mathbf{p}^{(t)})$ ▷ Prediction Step
  $\epsilon^{(t)} = \sum_n p_n^t |h^{(t)}(\mathbf{x}_n) - y_n|$
  $\beta^{(t)} = \frac{\epsilon^{(t)}}{1-\epsilon^{(t)}}$
  $w_n^{(t)} = w_n^{(t-1)} \beta^{1-|h^{(t)}(\mathbf{x}_n^{(t)} - y_n^{(t)}|}$ $\quad \forall n$ ▷ Weight Update Step
end for
$h_F(\mathbf{x}) = 1\left[\sum_{t=1}^T (log(\frac{1}{\beta^{(t)}})) h^{(t)}(\mathbf{x}) \geq \frac{1}{2} \sum_{t=1}^T log(\frac{1}{\beta^{(t)}})\right]$

Based on this type of feedback, the goal is to maximize the total reward the player receives over a horizon. Applications of the MAB problem include: A/B testing (figuring out which advertisement to display based on how the user interacts with the advertisement experience), robotic grasping (which way should the robot pick up an object), medical treatment (treating patient's health status as reward, advise treatment).

1.2.1 Explore-exploit algorithm

Assuming a stochastic bandit, how should the player pull the arms if they have only T rounds? The Explore-Exploit algorithm aims at answering this question. At a high-level, the algorithm proceeds in two phases:

1. **Explore Phase**: Pull each arm M times to estimate the mean reward.
2. **Exploit Phase**: Keep pulling the arm with the highest expected mean reward until T.

2 Summary

2.1 Multi-Armed Bandit: Explore-Exploit

The algorithm for the Explore-Exploit algorithm is shown in Alg. 2.

Theorem 1. Hoeffding's Inequality Consider a one-dimension distribution ν with expectation μ, where any sample $r \sim \nu$ is bounded such that $r \in [0, 1]$. Given T i.i.d samples, $\{r^{(t)}\}_{t=1}^T$, we have that for any ϵ:

$$p\left(\left|\sum_{t=1}^T \frac{r^{(t)}}{T} - \mu\right| \geq \epsilon\right) \leq 2e^{-2T\epsilon^2}$$

Intuitively, this inequality means that the estimate of the mean gets better with the more samples are available. We will use this inequality in the regret bound derivation for the Explore-Exploit algorithm.

Algorithm 2 Explore-Exploit

for $k = 1 \to K$ **do**
  for $m = 1 \to M$ **do**
    $a = k$
    `Receive`(r)
    $\hat{\mu}_k = \hat{\mu}_k + \frac{r}{M}$
  end for
end for
for $t = KM \to T$ **do**
  $a^{(t)} = \arg\max_{k'} \hat{\mu}_{k'}$
  `Receive`$(r^{(t)})$
end for

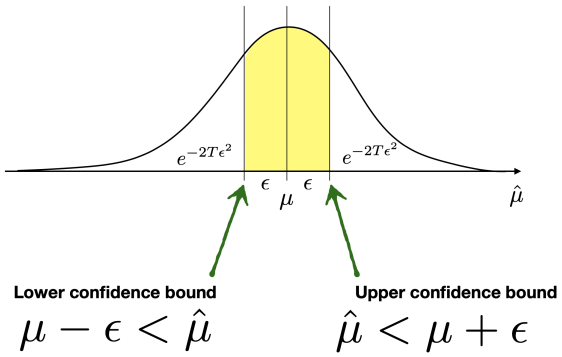

Figure 1: Illustrations of the definitions for the Lower Confidence Bound (LCB) and Upper Confidence Bound (UCB).

2.1.1 Regret Analysis

We will derive the regret bound in two steps: first for the exploration phase and then for the exploitation phase.

Exploration Phase For the exploration phase, the bound is $\mathcal{O}(KM)$ because it requires KM pulls and still gets no reward.

$$R_{\text{explore}} \leq \mathcal{O}(KM)$$

Exploitation Phase For the exploitation phase, we define the potential function using the fact that if we pulled the wrong arm (i.e. not the true best arm), that means our estimate of the best arm was wrong. Formally, our potential is defined as

$$\hat{\mu}_{\hat{k}} \geq \hat{\mu}_{k^*}$$

Now, we derive the upper and lower bounds of this potential.

Upper Bound For the upper bound, we can use the Upper Confidence Bound (UCB) (illustrated in Fig. 1):

$$\begin{aligned}\hat{\mu}_{\hat{k}} &\leq \mu_{\hat{k}} + \epsilon \\ &\leq \mu_{\hat{k}} + \sqrt{\frac{\log 2/\delta}{2M}}\end{aligned}$$

Lower Bound For the lower bound, we can use the Lower Confidence Bound (UCB) (illustrated in Fig. 1):

$$\begin{aligned}\hat{\mu}_{\hat{k}} &\geq \mu_{k^*} - \epsilon \\ &\geq \mu_{k^*} - \sqrt{\frac{\log 2/\delta}{2M}}\end{aligned}$$

Combining Lower and Upper Bounds We can combine these bounds as after some algebra derive the regret bound for the exploitation phase.

$$\begin{aligned}\mu_{\hat{k}} + \sqrt{\frac{\log(2/\delta)}{2M}} &\geq \mu_{k^*} - \sqrt{\frac{\log(2/\delta)}{2M}} \\ \mu_{k^*} - \mu_{\hat{k}} &\leq 2\sqrt{\frac{\log(2/\delta)}{2M}} \\ R_{\text{exploit}} = \sum_{t=KM+1}^{T} (\mu_{k^*}^{(t)} - \mu_{\hat{k}}^{(t)}) &\leq (T - KM) \cdot 2\sqrt{\frac{\log(2/\delta)}{2M}}\end{aligned}$$

Combining Exploration and Exploitation Bounds We now have expressions for the regret bounds for the explore and exploit phases. The sum gives us the final expression for the regret of the explore-exploit algorithm.

$$\begin{aligned}R_{\text{explore-exploit}} &= R_{\text{explore}} + R_{\text{exploit}} \\ &= KM + (2T - KM) \cdot \sqrt{\frac{1}{M}} \\ \leq KM + 2T \cdot \sqrt{\frac{1}{M}}\end{aligned}$$

Observe that the regret is linear in T, which is what we do not want. If we want our algorithm to be no-regret, we want the regret to grow sub-linearly. So, we can choose M in a way to make the regret sub-linear. We take the derivative of RHS and set it to zero.

$$\begin{aligned}0 &= \frac{d}{dM}\left\{KM + 2T \cdot \sqrt{\frac{1}{M}}\right\} \\ 0 &= K - TM^{-3/2} \\ M &= \left(\frac{T}{K}\right)^{2/3}\end{aligned}$$

Plugging this optimal M into our regret bound expression, we get:

$$\begin{aligned}
R &= R_{\text{explore}} + R_{\text{exploit}} \\
&\leq KM + 2T \cdot \sqrt{\frac{1}{M}} \\
&= K\left(\frac{T}{K}\right)^{2/3} + 2T\sqrt{\left(\frac{T}{K}\right)^{-2/3}} \\
&= K\left(\frac{T}{K}\right)^{2/3} + 2T\left(\frac{T}{K}\right)^{-1/3} \\
&= K^{1/3}T^{2/3} + 2T^{2/3}K^{1/3} \\
&= 3K^{1/3}T^{2/3} \\
&= \mathcal{O}(K^{1/3}T^{2/3})
\end{aligned}$$

Thus, the regret bound for the Explore-Exploit algorithm is $R = \mathcal{O}(K^{1/3}T^{2/3})$. Note that this is a sub-linear regret bound and that the total number of pulls T is assumed to be known.

2.2 Multi-Armed Bandit: Upper Confidence Bound

Upper Confidence Bound (UCB) is another algorithm to solve a context-free bandit problem. UCB is an optimistic learner where the arm $k(t)$ with the largest mean reward ($\hat{\mu}_k$) plus a confidence term is picked. The algorithm is detailed in Algo. 3.

Algorithm 3 Upper Confidence Bound (UCB)

for $t = 1 \rightarrow T$ **do**
  if $t \leq K$ **then**
    $k = t$ ▷ Initially pull each arm once (exploration)
  else
    $k = \arg\max_{k'} \left(\hat{\mu}_{k'} + \sqrt{\frac{\log(2T/\delta')}{2T_{k'}^{(t)}}}\right)$ ▷ upper confidence
  end if
  `RECEIVE`$(r^{(t)})$
  $T_k^{(t)} = T_k^{(t')} + 1$ ▷ update pull counter
  $\hat{\mu}_k = \frac{1}{T_k^{(t)}}\left(\hat{\mu}_k(T_k^{(t)} - 1) + r^{(t)}\right)$ ▷ update mean reward for k
end for

2.2.1 Understanding confidence term

The confidence term is obtained using Hoeffding's inequality and depends on the number of pulls of a particular arm $T_{k'}^{(t)}$, total pulls T and δ. So, as the game progresses and the number of pulls increase, the learner becomes more confident and the confidence term reduces.

Theorem 2. Union Bound The probability that at least one of the events happens is no greater than the sum of the probabilities of the individual events. Mathematically, $p(\bigcup_i x_i) \leq \sum_i p(x_i)$,

where x_i's are events.
Proof Sketch For two events x_1 and x_2, $p(x_1 \cup x_2) = p(x_1) + p(x_2) - p(x_1 \cap x_2)$
$p(x_1 \cup x_2) \leq p(x_1) + p(x_2)$
The same logic can be extended to arbitary n events.

Now using Hoeffding inequality (Theorem 1) and Union bound (Theorem 2),

$$p(\bigcup_{t=1}^{T} \left[|\hat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}} \right]) \leq \sum_{t=1}^{T} p\left(|\hat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}} \right) \leq T\delta = \delta'$$

Thus for the union event, the confidence interval becomes:

$$\epsilon' = \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}}$$

2.2.2 Regret Bound

For UCB, the regret bound comes out to $\mathcal{O}(\sqrt{KT})$. Next, we will derive the regret bound.

Base Inequality We first define the base inequality which is akin to potential function in some sense. If the learner selects a non-optimal arm $k \neq k^*$ where k^* is the optimal arm, we can say that the estimated mean plus the confidence term for that arm is higher than the same for the optimal arm (from prediction step, line 5 in Algo. 3). Mathematically, :

$$\hat{\mu}_k^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \hat{\mu}_{k^*}^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^*}^{(t)}}}$$

Upper Bound We wish to obtain a upper bound on the LHS of previous inequality. From Heoffding's inequality, $|\hat{\mu}^{(t)} - \mu| \leq \epsilon'$ or $\hat{\mu}^{(t)} \leq \epsilon' + \mu$ since we interested in upper bound. Using this, we have:

$$\hat{\mu}_k^{(t)} \leq \mu_k + \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}}$$

$$\mu_k + 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \hat{\mu}_k^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}}$$

Lower Bound We wish to obtain a upper bound on the RHS of the base inequality. From Heoffding's inequality, $|\hat{\mu}^{(t)} - \mu| \leq \epsilon'$ or $\hat{\mu}^{(t)} \geq \mu - \epsilon'$ since we interested in lower bound.

$$\hat{\mu}_{k^*}^{(t)} \geq \mu_{k^*} - \sqrt{\frac{\log(2T/\delta')}{2T_{k^*}^{(t)}}}$$

$$\hat{\mu}_{k^*}^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^*}^{(t)}}} \geq \mu_{k^*} - \sqrt{\frac{\log(2T/\delta')}{2T_{k^*}^{(t)}}} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^*}^{(t)}}}$$

$$\hat{\mu}_{k^*}^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^*}^{(t)}}} \geq \mu_{k^*}$$

Combining Bounds

$$\mu_k + 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \mu_{k^*}$$

$$\mu_{k^*} - \mu_k \leq 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}}$$

The above equation is for just one pull k. We are pulling an arm $k(t)$ at each time instant t. Thus,

$$\mathcal{R}_{UCB} = \sum_{t=1}^{T}(\mu_{k^*} - \mu_{k^{(t)}})$$

$$= \sum_{t=1}^{T} 2\sqrt{\frac{\log(2T/\delta')}{2T_{k^{(t)}}^{(t)}}}$$

$$= \sqrt{2\log(2T/\delta')}\sum_{t=1}^{T}\sqrt{\frac{1}{T_{k^{(t)}}^{(t)}}}$$

$$= \sqrt{2\log(2T/\delta')}\sum_{t=1}^{T}\sum_{j=1}^{K} 1[k^{(t)} = j]\sqrt{\frac{1}{T_j^{(t)}}}$$

$$= \sqrt{2\log(2T/\delta')}\sum_{j=1}^{K}\sum_{t=1}^{T} 1[k^{(t)} = j]\sqrt{\frac{1}{T_j^{(t)}}}$$

$$= \sqrt{2\log(2T/\delta')}\sum_{j=1}^{K}\sum_{t=1}^{T_j^{(T)}}\sqrt{\frac{1}{t}}$$

There are two mathematical inequalities required to derive the bound.

Fact 3. Summation Bound: $\sum_{t=1}^{T}\sqrt{\frac{1}{t}} \leq 2\sqrt{T}$

Proof. $\sum_{t=1}^{T}\sqrt{\frac{1}{t}} = 1 + \sum_{t=2}^{T}\sqrt{\frac{1}{t}} \leq 1 + \int_{t=1}^{T}\sqrt{\frac{1}{t}}dt$ *(as* $\sum_{t=n+1}^{T}\frac{1}{\sqrt{T}} \leq \int_{t=n}^{T}\frac{1}{\sqrt{T}}dt$*)*
$= 1 + 2\sqrt{T} - 2 = 2\sqrt{T} - 1 \leq 2\sqrt{T}$

Theorem 4. Jensen's Inequality If and only if f is a convex function, $f(\sum_n p_n x_n) \leq \sum_n p_n f(x_n)$. Equivalently, if f is concave, $\sum_n p_n f(x_n) \leq f(\sum_n p_n x_n)$.

Continuing with the derivation,

$$\mathcal{R}_{UCB} \leq \sqrt{2\log(2T/\delta')}\sum_{j=1}^{K}\sum_{t=1}^{T_j^{(T)}}\sqrt{\frac{1}{t}}$$

$$\leq \sqrt{2\log(2T/\delta')}\sum_{j=1}^{K} 2\sqrt{T_j^{(T)}} \qquad \text{(Using Fact 3)}$$

$$\leq \sqrt{8\log(2T/\delta')}K\left(\frac{1}{K}\sum_{j=1}^{K}\sqrt{T_j^{(T)}}\right)$$

$$\leq \sqrt{8\log(2T/\delta')}K\left(\sqrt{\frac{1}{K}\sum_{j=1}^{K}T_j^{(T)}}\right) \qquad \text{(Using Jensen's inequality)}$$

$$= \sqrt{8\log(2T/\delta')}K\left(\sqrt{\frac{T}{K}}\right)$$

$$= \sqrt{8\log(2T/\delta')}\sqrt{TK} \approx \mathcal{O}(\sqrt{TK})$$

In the slides, the constant term misses the factor of $\sqrt{8}$.

The regret bound for UCB is better than Explore-Exploit. This is because it starts exploiting earlier and continues exploring.

References

[1] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[2] R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.